*Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана  (национальный исследовательский университет)»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети

ДИСЦИПЛИНА Языки программирования для работы с большими данными

**Отчет**

**по лабораторной работе № 3**

**Вариант 5**

Студент гр. ИУ6-21М **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_**Войтов Д.Ю.\_\_\_\_

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2020

**ЗАДАНИЕ 1**

*Условие*

Определить класс «Матрица размерности (m x n)». Класс должен содержать несколько конструкторов. Объявить массив объектов. Передать объекты в метод, меняющий местами строки с максимальным и минимальным элементами k-го столбца. Создать метод, который изменяет i-ю матрицу путем возведения ее в квадрат.

*Решение*

Приведем описание класса Mtrx:

|  |
| --- |
| package com.company;  public class Mtrx { //матрИца  private int [][] elem;  public Mtrx(int [][] n){ //Конструктор с передачей массива  this.elem = n;  }  public Mtrx(int n, int m, int mx){ //Конструктор с рандомной генерацией  elem = new int[n][m];  for (int i = 0; i < n; i++){  for (int j = 0; j < m; j++){  this.elem[i][j] = (int) (Math.random() \* mx);  }  }  }  public void setNums(int [][] n){  this.elem = n;  }  public void Swap(int k){ //Переставляем строки с макс и мин в столбце k  int pos\_x = 0;  int pos\_n = 0;  int buf\_x = this.elem[0][k];  int buf\_n = this.elem[0][k];  for (int i = 0; i < this.elem.length; i++){  if (this.elem[i][k] > buf\_x){  buf\_x = this.elem[i][k];  pos\_x = i;  }  if (this.elem[i][k] < buf\_n){  buf\_n = this.elem[i][k];  pos\_n = i;  }  }  int [] buf\_ar = this.elem[pos\_x];  this.elem[pos\_x] = this.elem[pos\_n];  this.elem[pos\_n] = buf\_ar;  }  public void sqr(int i)  {  if ((i>this.elem.length)||(i>this.elem[i].length)){  System.out.println("i too big, bro :C");  }  else {  System.out.println("nice i, bro C:");  for (int k = 0; k < i; k++) {  for (int j=0; j<i; j++) {  this.elem[k][j]= (int) Math.pow(this.elem[k][j],2);  }  }  }  }  public void Show(){  for (int i = 0; i < this.elem.length; i++){  for (int j = 0; j < this.elem[i].length; j++){  System.out.print(this.elem[i][j] + " ");  }  System.out.println();  }  System.out.println();  } } |

**ЗАДАНИЕ 2**

*Условие*

Book: id, Название, Автор(ы), Издательство, Год издания, Количество страниц, Цена, Переплет. Создать массив объектов. Вывести: a) список книг заданного автора; b) список книг, выпущенных заданным издательством; c) список книг, выпущенных после заданного года.

*Решение*

Приведем описание требуемого класса:

|  |
| --- |
| package com.company;  import java.util.Arrays;  public class Book {  private int id;  private String name;  private String [] author;  private String publisher;  private int year;  private int pages;  private int price;  private String cover;   @Override  public String toString() {  return "Book{" +  "name='" + name + '\'' +  ", author=" + Arrays.toString(author) +  ", publisher='" + publisher + '\'' +  ", year=" + year +  ", pages=" + pages +  ", price=" + price +  ", cover='" + cover + '\'' +  '}';  }   public void setId(int nid) {  this.id=nid;  }   public void setName(String nname) {  this.name=nname;  }   public void setAuthor(String [] nauthor) {  this.author=nauthor;  }   public void setPublisher(String npublisher) {  this.publisher=npublisher;  }   public void setYear(int nyear) {  this.year=nyear;  }   public void setPages(int npages) {  this.pages=npages;  }   public void setPrice(int nprice) {  this.price=nprice;  }   public void setCover(String ncover) {  this.cover=ncover;  }   public int getId() {  return id;  }   public String getName() {  return name;  }   public String[] getAuthor() {  return author;  }   public String getPublisher() {  return publisher;  }   public int getYear() {  return year;  }   public int getPages() {  return pages;  }   public int getPrice() {  return price;  }   public String getCover() {  return cover;  }    public Book(int i, String n, String [] a, String p, int y, int pgs, int prc, String c){  this.id = i;  this.name = n;  this.author = a;  this.publisher = p;  this.year = y;  this.pages = pgs;  this.price = prc;  this.cover = c;  } |

Приведем код входной точки:

|  |
| --- |
| package com.company; import java.util.Scanner; public class Main {   public static void main(String[] args) {  Book books[] = new Book[5];  books[0] = new Book(0,"Mem book", new String[]{"Put", "In"}, "BlueHill", 2018, 100, 1200, "hard");  books[1] = new Book(1,"Old book", new String[]{"Put"}, "RedHill", 1980, 500, 1200, "ulta-soft");  books[2] = new Book(2,"Neo book", new String[]{"Mark Bookenberg"}, "WhiteHill", 2015, 130, 1200, "hard");  books[3] = new Book(3,"Book of Book", new String[]{"Big Ralph"}, "WhiteHill", 1993, 100, 1200, "soft");  books[4] = new Book(4,"Necronomicon?", new String[]{"Satan"}, "HellHill", 666, 666, 666, "human skin");   Scanner scan = new Scanner(System.in);  String author = scan.nextLine(); //Ввод автора для поиска  for (int i = 0; i < books.length; i++){ //  for (int j = 0; j < books[i].getAuthor().length; j++){  if (books[i].getAuthor()[j].equals(author)) {  System.out.println(books[i]);  }  }  }  System.out.println();   String publisher = scan.nextLine(); //Ввод издательства для поиска  for (int i = 0; i < books.length; i++){ //  if (books[i].getPublisher().equals(publisher)) {  System.out.println(books[i]);  }  }  System.out.println();   int year = scan.nextInt(); //Ввод года для поиска  for (int i = 0; i < books.length; i++){ //  if (books[i].getYear() > year) {  System.out.println(books[i]);  }  }   } } |

**ЗАДАНИЕ 3**

*Условие*

Создать объект класса Дом, используя классы Окно, Дверь. Методы: закрыть на ключ, вывести на консоль количество окон, дверей.

*Решение*

Приведем описание требуемого класса дома:

|  |
| --- |
| package com.company;  import java.util.ArrayList; import java.util.List;  public class House {  private List<Window> windows = new ArrayList<Window>();  private List<Door> doors = new ArrayList<Door>();  private String name;  private String state;   public House(String n, String s) {  this.name = n;  this.state = s;  }   public void show(){ //Вывод числа дверей и окон  System.out.println("This house is: " + this.name );  System.out.println("When you look on this, you see, what all doors is: " + this.state);  System.out.println("There:" +this.windows.size()+ " Windows");  System.out.println("There:" +this.doors.size()+ " Doors");  System.out.println();  }   public void add\_window(String s){  Window w = new Window(s);  windows.add(w);  }   public void add\_door(String s, String t){  Door d = new Door(s, t);  doors.add(d);  }  public void open(){  System.out.println("Outside doors opened");  System.out.println();  for (Door door : doors) {  if (door.getTitle().equals("Outside")) {  door.open();  }  }  this.state = "open";  }  public void close(){  System.out.println("Outside doors closed");  System.out.println();  for (Door door : doors) {  if (door.getTitle().equals("Outside")) {  door.close();  }  }  this.state = "closed";  }   public void lock(){  System.out.println("Outside doors now locked");  System.out.println();  for (Door door : doors) {  if (door.getTitle().equals("Outside")) {  door.lock();  }  }  this.state = "locked";  } |

Приведем описание класса-двери:

|  |
| --- |
| package com.company;  public class Door {  private String state; // open, closed, locked  private String title;   public String getState() {  return state;  }   public String getTitle() {  return title;  }   Door(String s, String t) {  this.state=s;  this.title=t;  }   public void open(){  this.state="open";  }   public void close(){  this.state="closed";  }   public void lock() {  this.state="closed by key";  } } |

Приведем описание класса-окна:

|  |
| --- |
| package com.company;  public class Window {  private String state; // Open or closed  Window(String s) {  this.state=s;  }  public void open(){  this.state="open";  }  public void close(){  this.state="closed";  }  public String getState()  {  return state;  } } |

**ЗАДАНИЕ 4**

*Условие*

Система Библиотека. Читатель оформляет Заказ на Книгу. Система осуществляет поиск в Каталоге. Библиотекарь выдает Читателю Книгу на абонемент или в читальный зал. При невозвращении Книги Читателем он может быть занесен Администратором в «черный список».

*Решение*

Приведем описание класса-книги:

|  |
| --- |
| package com.company;  public class Book {  private String name;  private int pages;  private String state;   public String getName() {  return name;  }   public void setState(String state) {  this.state = state;  }  public String getState()  {  return this.state;  }  public Book(String n, int p){  this.name = n;  this.pages = p;  this.state = "i"; // i - на скаде, r - в чит зале, a - абонемент  } } |

Приведем описание класса-пользователя:

|  |
| --- |
| package com.company;  public class Customer {  private String name;  private int InBlacklist;  public String getName(){  return name;  }  public int getInBlacklist() {  return InBlacklist;  }   public void setInBlacklist(int inBlacklist) {  this.InBlacklist = inBlacklist;  }   public Customer(String n){  this.name = n;  this.InBlacklist = 0;  }   public void Show(){  System.out.println(this.name + " : " + this.InBlacklist);  }  } |

Приведем описание класса-заказа:

|  |
| --- |
| package com.company;  public class Order {  private Customer cust;  private Book boo;  private String state; //u - up, c - closed  private int days;  public int getDays(){  return this.days;  }  public String getName(){  return this.cust.getName();  }  public String getBook(){  return this.boo.getName();  }  public Order(Customer cu, Book b, int d){  this.cust = cu;  this.boo = b;  this.state = "u";  if (d == 1) {  this.boo.setState("r"); //Если выдается на день - то в чит зал  }  else {  this.boo.setState("a"); //Если больше, то абонемент  }  this.days = d;  }  public void DayPassed(){  this.days -= 1;  if (this.days < 0) {  this.cust.setInBlacklist(-1);  }  }  public void Close(){  this.boo.setState("i");  if (this.days < 0) {  this.cust.setInBlacklist(0);  }  else {  this.cust.setInBlacklist(1);  }  } } |

**ВЫВОД**

В ходе выполнения данной работы средствами языка Java создавались классы, соответствующие заданию. Рассматривалось взаимодействие классов методом включения, использовалось переопределение конструкторов и автоматическая генерация методов получение и изменения параметров объекта.